

1
2 **In the Claims**

3 Claim 11 is currently amended.

4 Claims 1-4, 6-15, and 18-22 are pending and are listed below.

5
6 1. (Previously Presented) A method for generating a delta between
7 a first program binary and a second program binary, the method comprising the
8 steps of:

9 obtaining a first control flow graph (CFG) representation of the first binary
10 and obtaining a second CFG representation of the second binary;

11 comparing the first and second CFG representations to identify blocks
12 (nominally matched blocks) that match in the first and second CFG
13 representations, thereby identifying blocks (nominally unmatched blocks) in the
14 second CFG representation that do not match in the first CFG representation, the
15 comparing being based upon content of blocks being compared and augmented
16 local neighborhoods of blocks surrounding blocks being compared, wherein a
17 local neighborhood of a particular block consists of blocks neighboring that block
18 in a CFG representation, but less than all the blocks in that CFG representation,
19 and an augmented local neighborhood of that particular block consists that block's
20 local neighborhood plus a random sampling of blocks from a substantially larger
21 neighborhood of blocks surrounding that block, an augmented local neighborhood
22 in a CFG representation consisting of less than all the blocks in that CFG
23 representation;
24
25

1 determining edit-operations that merges the unmatched blocks into the first
2 CFG representation so that first CFG representation is substantially identical to the
3 second CFG representation;

4 producing a delta comprising the unmatched blocks and the edit-operations.

5
6 2. (Original) A method for transmitting a delta comprising:
7 a method for generating a delta as recited in claim 1;
8 transmitting the delta over a network.

9
10 3. (Original) A method for patching a copy of the first program
11 binary, the method comprising:
12 a method for generating a delta as recited in claim 1;
13 patching the copy of the first program binary so that the copy is
14 substantially identical to the second program binary, wherein the delta guides such
15 patching.

16
17 4. (Previously Presented) A method as recited in claim 1, wherein
18 a local neighborhood of a particular block consists of those blocks immediately
19 adjacent that block.

20
21 5. (Canceled)

1 6. (Original) A computer-readable medium having embodied
2 thereon a data structure, comprising a delta generated in accordance with the steps
3 recited in claim 1.

4
5 7. (Original) A computer-readable medium having computer-
6 executable instructions that, when executed by a computer, performs the method
7 as recited in claim 1.

8
9 8. (Original) A method for matching blocks between a first control
10 flow graph (CFG) representation of a portion of a first program and a second CFG
11 representation of a portion of a second program, the method comprising:

12 matching blocks between the first and second CFG representations based
13 upon the content of the blocks;

14 detecting outliers, wherein outliers are blocks in the first CFG
15 representation that do not match any block in the second CFG representation
16 during the matching step;

17 computing a neighborhood of each block in the first and second CFG
18 representations by performing a breadth first traversal;

19 removing the outliers from each neighborhood.

20
21 9. (Previously Presented) A method as recited in claim 8 further
22 comprising:

23 computing labels for each block in first and second CFG representations
24 based upon content of a block;

25

1 for each neighborhood computed in the computing step, forming a "d-
2 label" for each block in a neighborhood based upon labels of the blocks within the
3 neighborhood;

4 attempting to match blocks between first and second CFG representations
5 by comparing the d-labels of the blocks.

6
7 10. (Original) A computer-readable medium having computer-
8 executable instructions that, when executed by a computer, performs the method
9 as recited in claim 8.

10
11 11. (Currently Amended) A method for matching procedures
12 between a first control flow graph (CFG) representation of a portion of a first
13 program and a second CFG representation of a portion of a second program,
14 wherein a procedure comprises multiple blocks in a CFG representation, the
15 method comprising:

16 computing a procedure-match-criterion for each procedure in the second
17 CFG representation, where the procedure-match-criterion for a procedure in the
18 second CFG representation represents the number of matching blocks between that
19 procedure and a specified procedure in the first CFG representation;

20 matching procedures in the second CFG representation with the specified
21 procedure in the first CFG representation based upon the procedure-match-criteria
22 for the procedures in the second CFG representation[.];

23 attempting to match blocks in the procedure in the second CFG
24 representation with blocks in the specified procedure in the first CFG
25 representation.

1
2 12. (Original) A computer-readable medium having computer-
3 executable instructions that, when executed by a computer, performs the method
4 as recited in claim 11.
5

6 13. (Previously Presented) A method for matching of blocks in a
7 procedure of a first control flow graph (CFG) representation of a portion of a first
8 program between an ostensibly matching procedure of a second CFG
9 representation of a portion of a second program, the method comprising:

10 matching blocks between the first and second CFG representations based
11 upon the content of the blocks;

12 computing successively smaller neighborhoods of each block in the first
13 and second CFG representations via breadth first traversals;

14 for each neighborhood computed in the computing step, forming a "d-
15 label" for each block in a neighborhood based upon labels of the blocks within the
16 neighborhood;

17 attempting to match blocks between first and second CFG representations
18 by comparing the d-labels of the blocks.
19

20 14. (Original) A method as recited in claim 13, wherein at least one
21 neighborhood computed in the computing steps is augmented with a random
22 sampling of blocks in the complete representation of the neighborhood.
23
24
25

1 15. (Original) A computer-readable medium having computer-
2 executable instructions that, when executed by a computer, performs the method
3 as recited in claim 13.

4
5 16. (Not Entered)

6
7 17. (Not Entered)

8
9 18. (Previously Presented) A patch data structure generated in
10 accordance with the following acts:

11 providing a server computer in a communications with a communications
12 network;

13 receiving input from a client computer by way of the communications
14 network, the input providing a parameter indicative of a request for upgrading a
15 copy of a first program binary to a match a second program binary;

16 retrieving a delta between a first program binary and the second program
17 binary, wherein computing such delta comprises the steps of:

18 a) obtaining a first control flow graph (CFG) representation of the first
19 binary and obtaining a second CFG representation of the second
20 binary;

21 b) comparing the first and second CFG representations to identify
22 blocks (nominally matched blocks) that match in the first and second
23 CFG representations, thereby identifying blocks (nominally
24 unmatched blocks) in the second CFG representation that do not
25 match in the first CFG representation, the comparing being based

1 upon content of blocks being compared and augmented local
2 neighborhoods of blocks surrounding blocks being compared,
3 wherein a local neighborhood of a particular block consists of blocks
4 neighboring that block in a CFG representation, but less than all the
5 blocks in that CFG representation, and an augmented local
6 neighborhood of that particular block consists that block's local
7 neighborhood plus a random sampling of blocks from a substantially
8 larger neighborhood of blocks surrounding that block, an augmented
9 local neighborhood in a CFG representation consisting of less than
10 all the blocks in that CFG representation;

- 11 c) determining edit-operations that merges the unmatched blocks into
12 the first CFG representation so that first CFG representation is
13 substantially identical to the second CFG representation;
14 d) producing a delta comprising the unmatched blocks and the edit-
15 operations;

16 generating the patch data structure as a function of the delta.

17
18 19. (Original) A method for transmitting a patch data structure
19 comprising transmitting a patch data structure as recited in claim 18 over a
20 communications network.
21
22
23
24
25

1 20. (Original) A method for patching a copy of the first program
2 binary at a client computer, the method comprising patching the copy of the first
3 program binary so that the copy is substantially identical to the second program
4 binary, wherein a delta in a patch data structure as recited in claim 18 guides such
5 patching.

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

1 21. (Previously Presented) A delta-generator system, comprising:
2 a comparator that is configured to compare a first control flow graph (CFG)
3 representation of a first program binary and a second CFG representation of the
4 second program binary for identifying blocks (nominally matched blocks) that
5 match in the first and second CFG representations, thereby identifying blocks
6 (nominally unmatched blocks) in the second CFG representation that do not match
7 in the first CFG representation, the comparing being based upon content of blocks
8 being compared and augmented local neighborhoods of blocks surrounding blocks
9 being compared, wherein a local neighborhood of a particular block consists of
10 blocks neighboring that block in a CFG representation, but less than all the blocks
11 in that CFG representation, and an augmented local neighborhood of that
12 particular block consists that block's local neighborhood plus a random sampling
13 of blocks from a substantially larger neighborhood of blocks surrounding that
14 block, an augmented local neighborhood in a CFG representation consisting of
15 less than all the blocks in that CFG representation;
16 an edit-op determiner configured to determine the edit-operations that
17 merges the unmatched blocks into the first CFG representation so that first CFG
18 representation is substantially identical to the second CFG representation;
19 an output sub-system that is configured to produce a delta comprising the
20 unmatched blocks and the edit-operations.
21

22 22. (Original) A computer-readable medium having embodied thereon a
23 data structure comprising a delta produced by the system as recited in claim 21.
24
25